Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки информации и управления»

Отчет

Лабораторная работа № 2

По курсу «Технологии машинного обучения»

ИСПОЛНИТЕЛЬ:

Горбатенко И.А.

Группа ИУ5-64

_____

"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

_____

"__"_____2020 г.

Москва 2020

Лабораторная работа №2 по курсу "Технологии машинного обучения"

Горбатнко И.А. ИУ5-64

Цель лабораторной работы: изучение библиотеки обработки данных Pandas.

Задание:

Выполните первое демонстрационное задание "demo assignment" под названием "Exploratory data analysis with Pandas" со страницы курса
https://mlcourse.ai/assignments (https://mlcourse.ai/assignments)

Выполнение:

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
data = pd.read_csv('adult.data.csv')
data.head()
```

Out[2]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female

Количество мужчин и женщин

In [3]:
```
data['sex'].value_counts()
```

Out[3]:
```
Male      21790
Female    10771
Name: sex, dtype: int64
```

Средний возраст женщин

In [4]:
```
data.loc[data['sex'] == 'Female', 'age'].mean()
```

Out[4]:
```
36.85823043357163
```

Доля граждан Германии

In [5]:
```
float((data['native-country'] == 'Germany').sum()) / data.shape[0]
```

Out[5]:
```
0.004207487485028101
```

Среднее значение и стандартное отклонение возраста людей из двух категорий: тех, кто получал более 50 тысяч в год и тех, кто получал менее 50 тысяч в год

In [7]:
```
ages1 = data.loc[data['salary'] == '>50K', 'age']
ages2 = data.loc[data['salary'] == '<=50K', 'age']
print("Средний возраст получающих более 50 тысяч: {0} +- {1} лет, получающих менее
    round(ages1.mean()), round(ages1.std(), 1),
    round(ages2.mean()), round(ages2.std(), 1)))
```

```
Средний возраст получающих более 50 тысяч: 44 +- 10.5 лет, получающих менее 50 ты
сяч: - 37 +- 14.0 years.
```

Правда ли, что люди, получающие более 50 тысяч в год, имеют по крайней мере среднее образование?

In [8]:
```
data.loc[data['salary'] == '>50K', 'education'].unique()
```

Out[8]:
```
array(['HS-grad', 'Masters', 'Bachelors', 'Some-college', 'Assoc-voc',
       'Doctorate', 'Prof-school', 'Assoc-acdm', '7th-8th', '12th',
       '10th', '11th', '9th', '5th-6th', '1st-4th'], dtype=object)
```

Как видно, ответ "неправда"

Статистика возрастов для каждой расы и пола, максимальный возраст мужчин расы Amer-Indian-Eskimo.

In [14]:
```
for (race, sex), sub_df in data.groupby(['race', 'sex']):
    print("Раса: {0}, Пол: {1}".format(race, sex))
    print(sub_df['age'].describe())
```

```
Раса: Amer-Indian-Eskimo, Пол: Female
count    119.000000
mean      37.117647
std       13.114991
min       17.000000
25%       27.000000
50%       36.000000
75%       46.000000
max       80.000000
Name: age, dtype: float64
Раса: Amer-Indian-Eskimo, Пол: Male
count    192.000000
mean      37.208333
std       12.049563
min       17.000000
25%       28.000000
50%       35.000000
75%       45.000000
max       82.000000
Name: age, dtype: float64
Раса: Asian-Pac-Islander, Пол: Female
count    346.000000
mean      35.089595
std       12.300845
min       17.000000
25%       25.000000
50%       33.000000
75%       43.750000
max       75.000000
Name: age, dtype: float64
Раса: Asian-Pac-Islander, Пол: Male
count    693.000000
mean      39.073593
std       12.883944
min       18.000000
25%       29.000000
50%       37.000000
75%       46.000000
max       90.000000
Name: age, dtype: float64
Раса: Black, Пол: Female
count    1555.000000
mean       37.854019
std        12.637197
min        17.000000
25%        28.000000
50%        37.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Раса: Black, Пол: Male
count    1569.000000
mean       37.682600
```

```
std        12.882612
min        17.000000
25%        27.000000
50%        36.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Раса: Other, Пол: Female
count    109.000000
mean      31.678899
std       11.631599
min       17.000000
25%       23.000000
50%       29.000000
75%       39.000000
max       74.000000
Name: age, dtype: float64
Раса: Other, Пол: Male
count    162.000000
mean      34.654321
std       11.355531
min       17.000000
25%       26.000000
50%       32.000000
75%       42.000000
max       77.000000
Name: age, dtype: float64
Раса: White, Пол: Female
count    8642.000000
mean       36.811618
std        14.329093
min        17.000000
25%        25.000000
50%        35.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Раса: White, Пол: Male
count    19174.000000
mean        39.652498
std         13.436029
min         17.000000
25%         29.000000
50%         38.000000
75%         49.000000
max         90.000000
Name: age, dtype: float64
```

Среди кого больше доля тех, кто зарабатывает больше 50 тыс в год: среди женатых мужчин или одиноких? (Женатые - те, у кого атрибут martial-status начинается с "Married")

In [10]:
```
data.loc[(data['sex'] == 'Male') &
     (data['marital-status'].isin(['Never-married',
                                   'Separated',
                                   'Divorced',
                                   'Widowed'])), 'salary'].value_counts()
```

Out[10]:
```
<=50K    7552
>50K      697
Name: salary, dtype: int64
```

In [11]:
```
data.loc[(data['sex'] == 'Male') &
     (data['marital-status'].str.startswith('Married')), 'salary'].value_co
```

Out[11]:
```
<=50K    7576
>50K     5965
Name: salary, dtype: int64
```

Как видим, в среднем женатые мужчины зарабатывают больше

Максимальное количество часов, которые человек работает в неделю. Количество людей, работающих такое количество часов. Процент тех, кто много зарабатывает среди них

In [17]:
```
max_load = data['hours-per-week'].max()
print("Максимальное количество часов в неделю - {0} ч".format(max_load))

num_workaholics = data[data['hours-per-week'] == max_load].shape[0]
print("Количесто таких же работяг - {0}".format(num_workaholics))

rich_share = float(data[(data['hours-per-week'] == max_load)
                 & (data['salary'] == '>50K')].shape[0]) / num_workaholics
print("Доля из них много зарабатывающих - {0}%".format(int(100 * rich_share)))
```

```
Максимальное количество часов в неделю - 99 ч
Количесто таких же работяг - 85
Доля из них много зарабатывающих - 29%
```

Среднее время работы тех, кто зарабатывает мало и много для каждой страны

In [18]:
```
pd.crosstab(data['native-country'], data['salary'],
            values=data['hours-per-week'], aggfunc=np.mean).T
```

Out[18]:

native-country	?	Cambodia	Canada	China	Columbia	Cuba	Dominican-Republic	Ecuador
salary								
<=50K	40.164760	41.416667	37.914634	37.381818	38.684211	37.985714	42.338235	38.041667
>50K	45.547945	40.000000	45.641026	38.900000	50.000000	42.440000	47.000000	48.750000

2 rows × 42 columns

Выводы: в Камбодже лучше не жить:)

In [ ]:
```
```